M6 - Final presentation
Group 3

Advait Dixit
Jiaqiang Ye Zhu
Manel Guzman
Albert Barreiro

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Universitat Autònoma de Barcelona

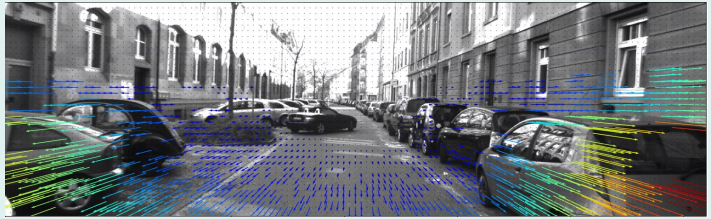
Universitat Pompeu Fabra Barcelona

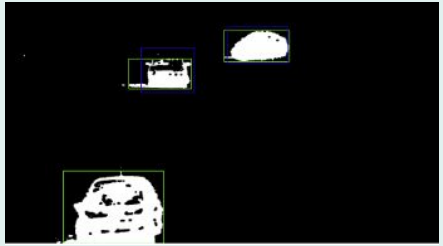
Universitat Oberta de Catalunya

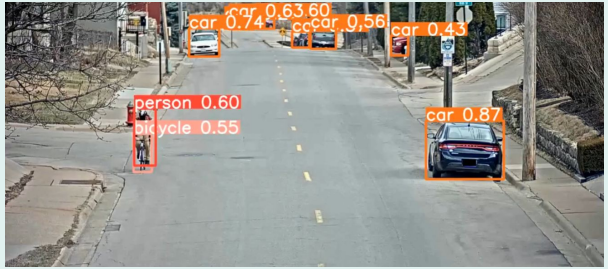
Table of contents

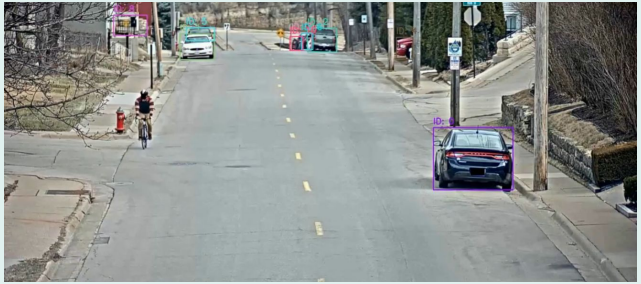
1. Summary of previous weeks
2. MTSC - Multi-target single camera tracking
3. MTMC - Multi-target multi-camera tracking
4. Discussions

Summary of 1-4 weeks



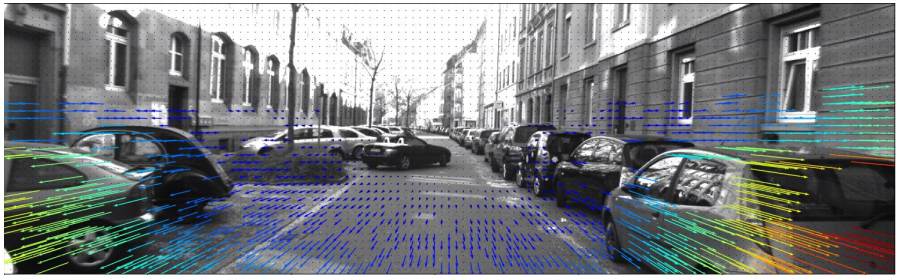


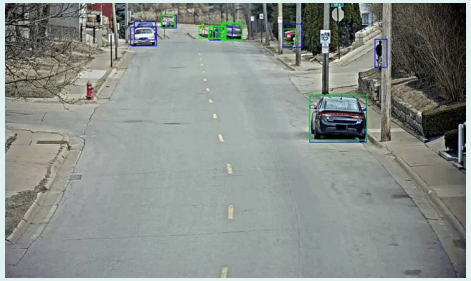


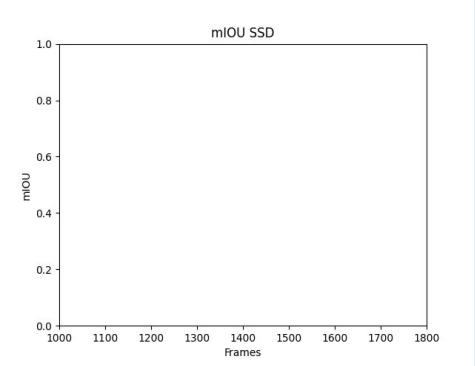


Week 1

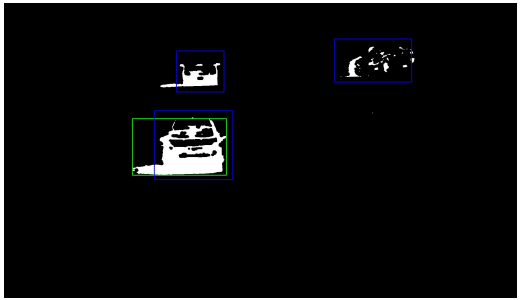
- Optical flow
- Evaluation of the metrics (mIOU, mAP)



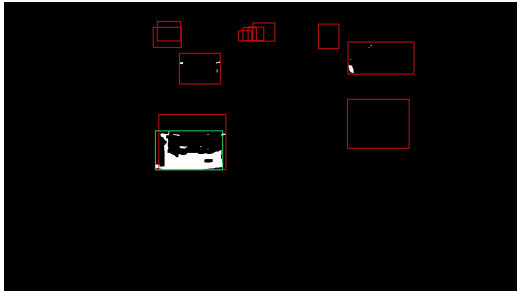




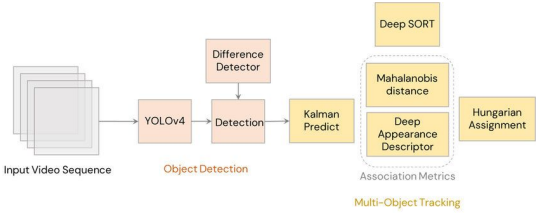
Week 2



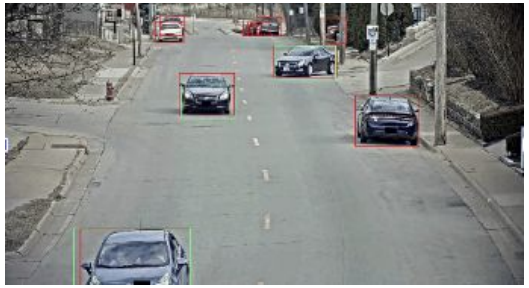
Gaussian modelling



Adaptive method



Comparison with: CNT, CSOC, LSBP, MOG/MOG2 and DeepSORT



XYZ(XZ)

Week 3

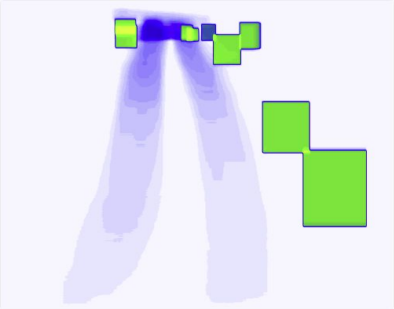
roboflow

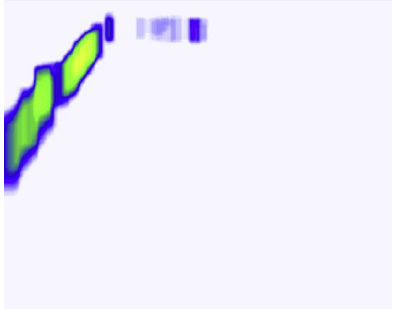
Detection by Off-the-Shelf[1] & Fine-tuning

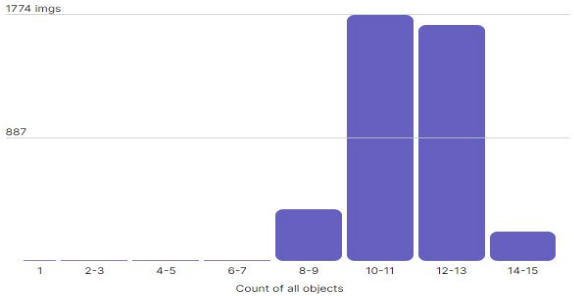
- Optical flow
- Annotations from scratch
- Evaluation of the metrics (**mIOU, mAP**)

Tracking by Maximum Overlap & Kalman Filter

Annotate:
More than 14 hours annotating

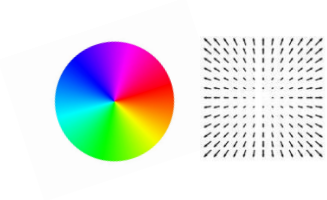


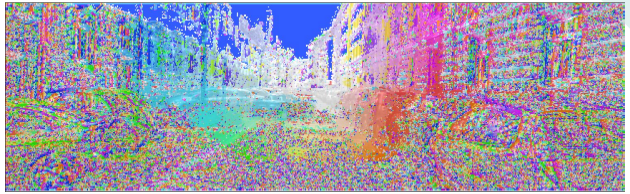




1. https://github.com/ultralytics

Week 4





Optical flow with block size of 4

Optical flow with minimum MSEN (block size of 32)

- Optical Flow with block matching
- Off-the-shelf Optical Flow
- Object tracking with Optical Flow
- MTSC Tracking

000045_10

000045_11

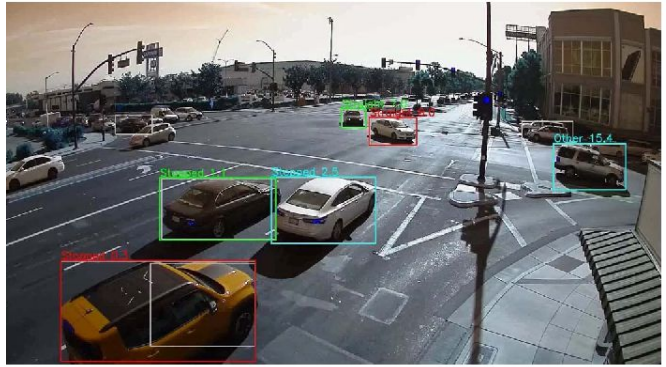
PWC-NET

Lukas-Kanade

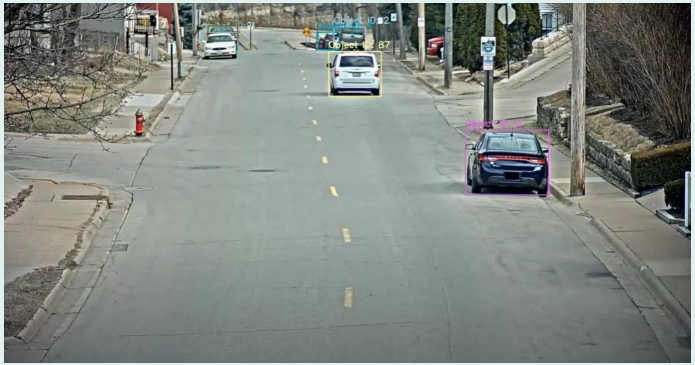
Pyflow

Week 5 - Motivation

The **AI City Challenge** is a competition aimed at using artificial intelligence to improve urban mobility and transportation systems. In this specific task, the objective is to develop AI algorithms that can accurately detect and track vehicles across multiple cameras, both at a single intersection and across multiple intersections in a mid-sized city.



Multi-target single-camera (MTSC) tracking



Car detection → Tracking by overlap → Remove Parked cars

Multi-target single-camera (MTSC) tracking

We have tested three different backbones for the object detection problem.

- The Mask R-CNN
- The SSD 512
- The YOLO3

Two different tracking techniques applied:

- Tracking by overlapping
- Tracking using the kalman filter

On those techniques we have applied two improvement approaches

- Refine the box: Do a box refinement based on the previous boxes detected[1]
- Remove parked cars detection based on the distance traveled.

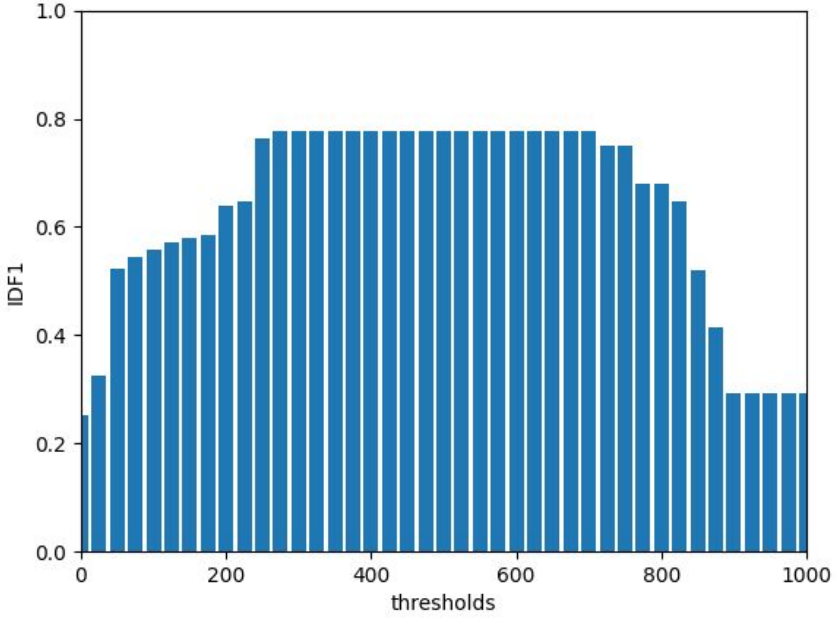
1. Borrowed from Team 2 2020

Multi-target single-camera (MTSC): Remove park cars algorithm

In order to remove parked cars we have check if a car has moved more than a threshold.

The **threshold** has been set by grid search from 0 to 1000 with a step size of 25. We set **YOLO V3** and using the camera 010 from sequence 3 to perform the detections.

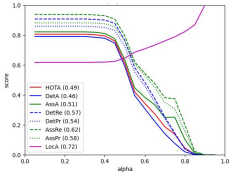
We have selected **550**.



Multi-target single-camera (MTSC) tracking

	IDF1 (SEQ 3)						
Camera	**c10**	**c11**	**c12**	**c13**	**c14**	**c15**	**Average**
Overlap,Mask R-CNN	0.609	0.270	0.611	0.472	0.785	0.202	0.491
Overlap,SSD 512	0.895	0.692	0.631	0.872	0.792	0.0	0.647
Overlap,YOLO	0.749	**0.699**	0.672	**0.873**	0.771	0.221	**0.664**
Kalman,Mask R-CNN	0.584	0.371	**0.732**	0.732	0.681	0.200	0.550
Kalman,SSD 512	**0.896**	0.482	0.721	0.723	**0.842**	0.0	0.610
Kalman,YOLO	**0.896**	0.649	0.642	0.831	0.751	**0.241**	**0.664**

Multi-target single-camera (MTSC) tracking

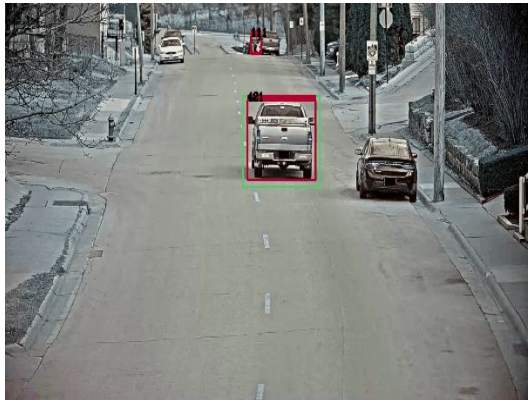


	HOTA (SEQ 3)						
Camera	**c10**	**c11**	**c12**	**c13**	**c14**	**c15**	**Average**
Overlap,Mask R-CNN	0.552	0.221	0.409	0.336	**0.654**	**1.0**	0.529
Overlap,SSD 512	0.693	0.269	0.457	**0.716**	0.651	0	0.464
Overlap,YOLO	0.693	**0.332**	0.455	0.707	0.635	**1.0**	**0.637**
Kalman,Mask R-CNN	0.552	0.242	0.516	0.373	0.471	**1.0**	0.526
Kalman,SSD 512	0.706	0.214	**0.547**	0.515	0.632	0	0.436
Kalman,YOLO	**0.729**	0.256	0.494	0.675	0.524	0.941	0.603

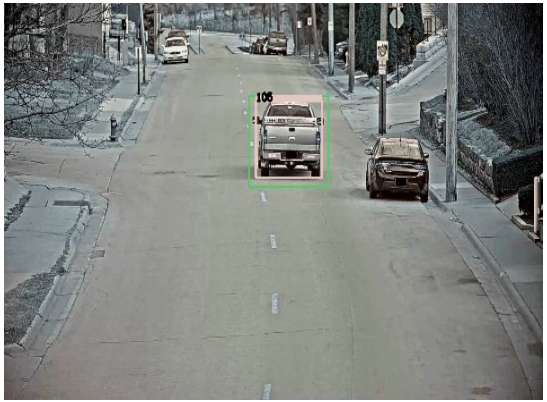
Multi-target single-camera (MTSC) Qualitative results: backbone comparison

Ground truth

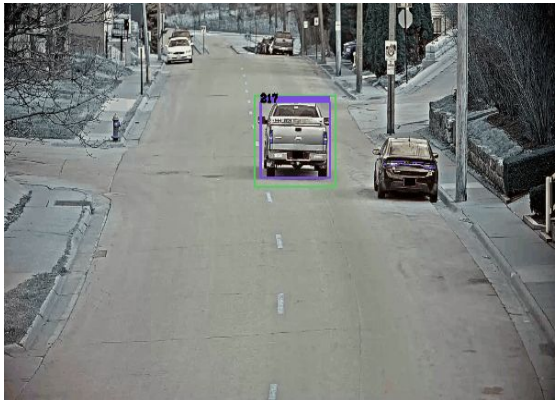
Prediction



Mask R-CNN

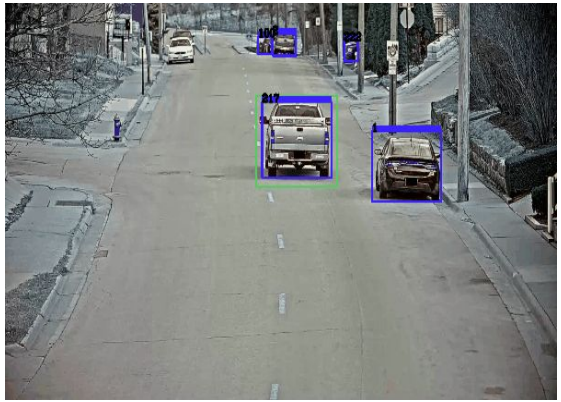


SSD 512

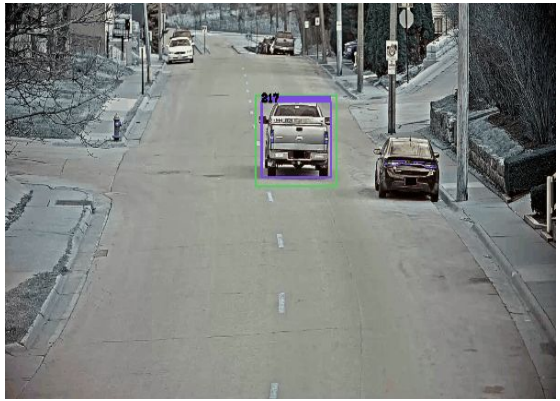


YOLO V3

Multi-target single-camera (MTSC) Qualitative results: Remove park cars



Park cars not removed



Park cars removed

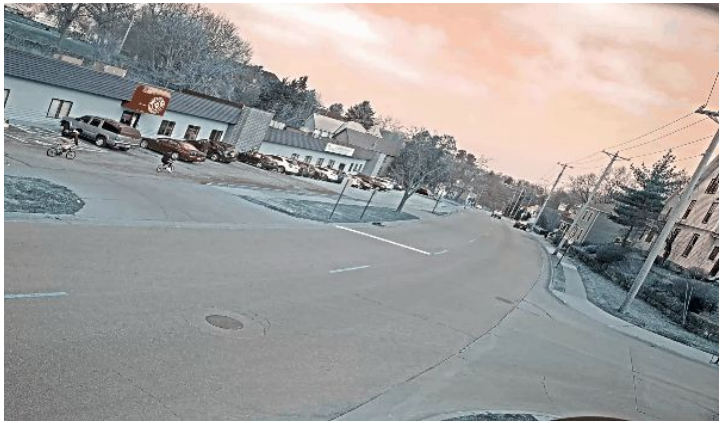
Ground truth

Prediction

*Using tracking by overlap and YOLO v3 as backbone

Multi-target single-camera (MTSC): Qualitative results

Cars in cam 15



Cars in cam 11

Multi-target multi-camera (MTMC) tracking

Tracking

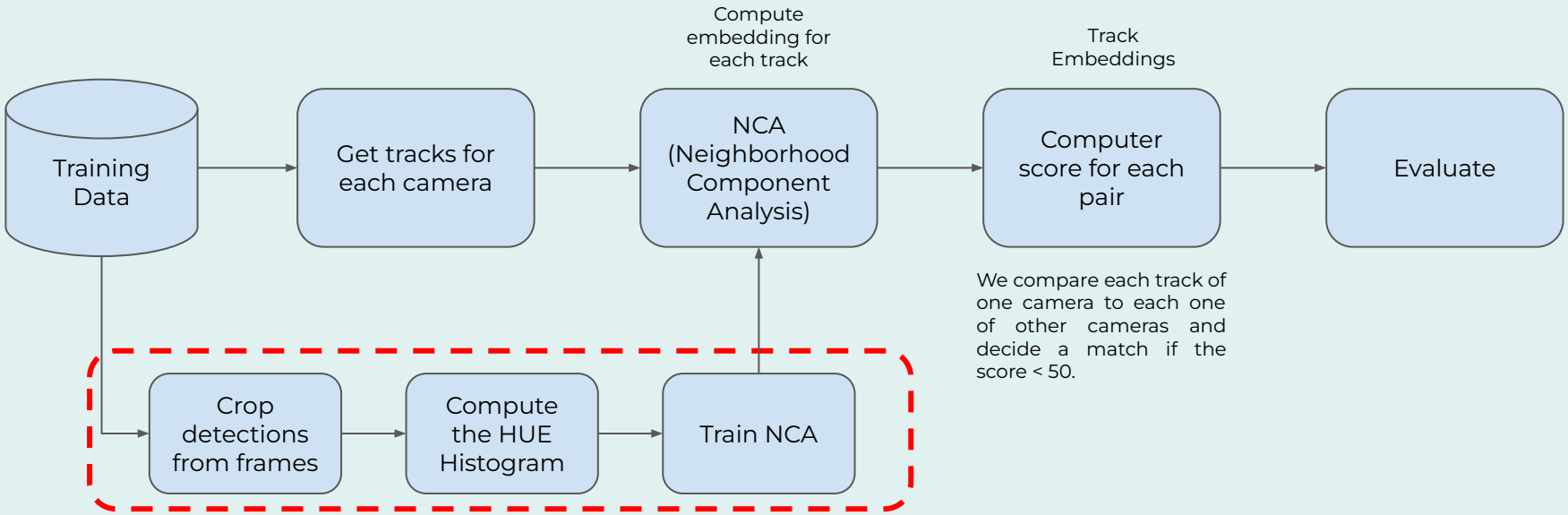


Diagram idea from Team 8 2021

W5.T2: Multi-target multi-camera (MTMC)

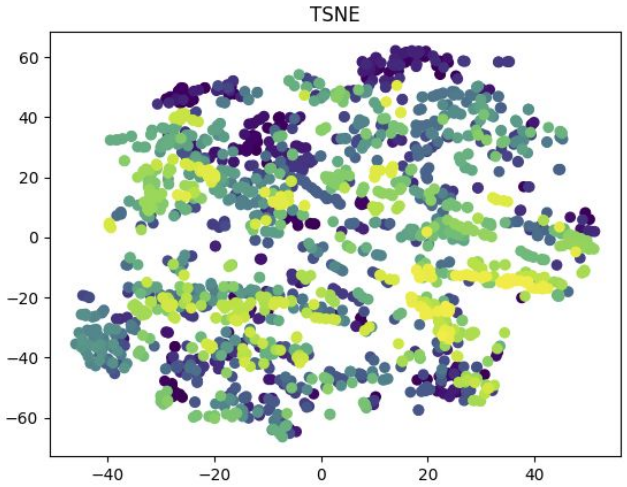
In the previous method, we use the Neighbourhood Components Analysis (NCA) algorithm which aims to learn a distance metric by finding a linear transformation of input data. The object will be cropped according to its bounding box then NCA will be trained on the histogram feature vector computed on this patch.

We use the pre-trained detection provided from the dataset

Training sequences: [S01, S04], validation: S03

Histogram: Hue histogram with Hue value

NCA parameters: init='pca', n_components=400, max_iter=1500



S01 C01 - Each color represents a different car

W5.T2: Multi-target multi-camera (MTMC) tracking SEQ 1

Sequences	Detectors	IDF1	IDR	IDP	Precision	Recall
Seq 1	Mask RCNN	31.35	28.91	**34.25**	44.72	**52.97**
	SSD	**37.53**	**44.36**	32.53	**69.64**	51.06
	YOLO	37.00	41.00	33.72	62.89	51.73

W5.T2: Multi-target multi-camera (MTMC) tracking SEQ 3

Sequences	Detectors	IDF1	IDR	IDP	Precision	Recall
Seq 3	Mask RCNN	45.20	38.35	**55.02**	67.85	**97.32**
	SSD	**49.42**	45.05	54.72	77.97	94.72
	YOLO	48.79	**46.61**	51.18	**84.86**	93.16

W5.T2: Multi-target multi-camera (MTMC) tracking SEQ 4

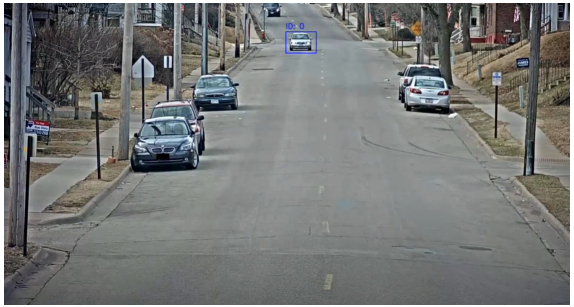
Sequences	Detectors	IDF1	IDR	IDP	Precision	Recall
Seq 4	Mask RCNN	47.73	44.08	**57.74**	63.73	**90.45**
	SSD	**48.66**	**51.49**	49.92	**76.98**	80.98
	YOLO	50.74	54.27	52.31	77.86	82.49

Multi-target single-camera (MTMC): Qualitative results

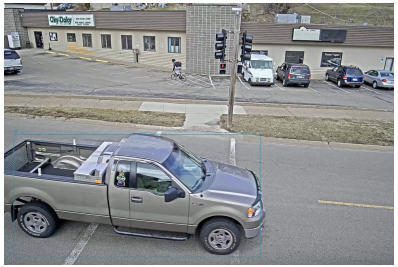
C10

C11

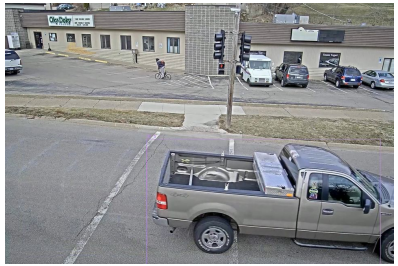
C14

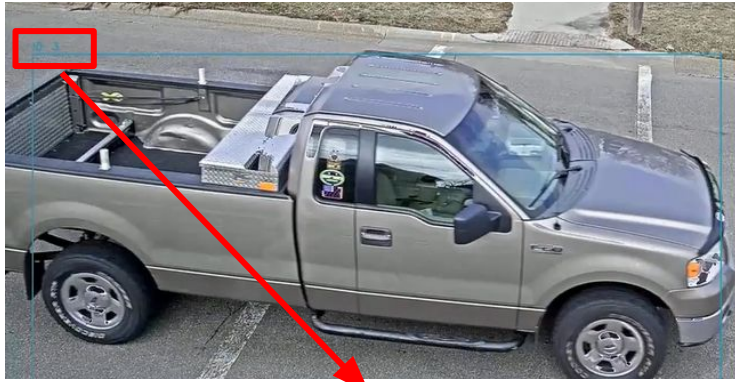


Multi-target single-camera (MTMC): Qualitative results



5 Frames later





ID: 3

ID: 5

Multi-target single-camera (MTMC): Qualitative results

C10

C11

Misclassified as car 2

Final conclusions

- Quantitative and qualitative **results of Single/Multi Camera are satisfactory**.
- **Post-processing** in Single Camera is very important in order to obtain accurate results.
- Tried several detectors and with a wide range of performance and we conclude that **YOLO** is the one that has **better compromise between speed and accuracy**.
- Dealt with **tracking issues due to occlusions** and **the ground truth**.
- Used **offline methods to obtain a higher accuracy** but the method can **easily** be **transformed** to **online**.